***Digital CCP Documentation – 2025***

***About DCCP:***

***DCCP is a “Document Generation Platform” which is used to streamline the documentation processes that are involved in healthcare workflows. The main objective of this project is to educate the patients about what to do and don’t do during pre-op and post-op phases of their surgery by creating customized documents and access the patient health conditions by using real-time customizable surveys to diagnose them better.***

***And The core functional part of this project is to create various type of documents including Structured, Unstructured, Survey documents to enhance the patient diagnoses and patient education towards their health.***

***Modules of DCCP:***

***As this product is seperated into 9 core modules with each module is handling unique functionality. Let me list down the all core modules of DCCP.***

***1) Dashboard***

***2) Library***

***3) Documents***

***4) Gallery***

***5) Report***

***6) Admin***

***7) My Profile***

***8) Dictionary***

***9) Inbox***

***However, In this document I’m gonna explain “DataDictionary and Inbox” module in detail, Let me explain all the 9 modules a little, so that you can have an idea of what each module is responible for.***

***1) Dashboard:***

***This is the initial screen that you will be seeing after you have sucessfully login to our system. As it name suggests it act as a insight visualization page, where we get some analytics and insights about the user’s current activity inside the application. It mainly contains 3 core components namely,***

***1.1) Recent Documents – Which represents the given users last four used documents and you can also apply filter to these documents by created\_date, document’s status.***

***1.2) My Document Summary – A summary which represents the overall documents count created by the user and grouped based on the status of the document.***

***1.3) Document Code Summary – A summary about the number of library documents and***

***their respective doccode’s count for each businessunit which has atleast one document in library.***

***2) Library:***

***This module is for displaying and managing documents which are in library status. It also provides way to download the each document in various formats including HTML, Epub formats. And this module consists of only 3 components.***

***2.1) Library Document List – This component is for displaying all the library documents present inside the current user’s client. You can view the document’s content by clicking on the view button present in the document list.***

***2.2) Filter Documents – This component is for adding a filter functionality to the library documents that we will be displaying. And the parameters that we use for filtering are Businessunit, Document type, Document Status, Document’s Created\_Date and Masterdocument name.***

***2.3) Download Documents – This component provides functionalities like downloading the document’s details (name, doctitle, doccode) in both csv & excel files and you can also download multiple unstructured documents in HTML, Epub formats. And each and every other documents can be downloaded in various formats including Pdf, HTML, Epubs.***

***3) Documents:***

***This module holds the same components and functionalities like Library component. However the main difference between these two modules are Library module is for displaying and managing “Documents in Library Status”. On the other hand Documents module is for displaying and managing “Documents in Other Statuses Except Library” which are non library documents.***

***This module also consists of 3 components.***

***3.1) Document List – This component is for displaying all the documents which are not in the library status of the current user’s client. You can either view or edit documents which are present in the document list.***

***3.2) Filter Documents – This component is for adding a filter functionality to the documents that we will be displaying. And the parameters that we use for filtering are Businessunit, Document type, Document Status, Document’s Created\_Date and Masterdocument name.***

***3.3) Download Documents – This component provides functionalities like downloading the document’s details (name, doctitle, doccode) in both csv & excel files and you can also download multiple unstructured documents in HTML, Epub formats. And each and every other documents can be downloaded in various formats including Pdf, HTML, Epubs.***

***4) Gallery:***

***This module is responsible for displaying and managing images and videos of our system. And it acts as a Gallery that we use on our phones, will consists the media files of our system. The main usecase of this module is to manage the media files so that we can use the same inside our documents especially for unstructured documents. This module is mainly consists of three components.***

***4.1) Display Media files – This component is responsible for displaying all the media files from our system filtered by the current user’s client details. And it also have option to preview and delete specific media files.***

***4.2) Upload Media files – This component is responible for uploading either single or multiple media files with specific categories including (Logo, Others, Videos, Signatures).***

***4.3) Filter Media files – This component is responible for filtering the media files based on their name and the category they are from including (All, Logo, Others, Videos, Signatures).***

***5) Report:***

***This module is acts as a report and insights section of our system by providing insights like Audit Report, User vs Comment, Businessunit vs Comment and so on. In general it contains 4 tabs with each of them having unique functionalities. So lets see what each component (tab) consists.***

***5.1) Audit Report – This component is responsible for displaying the user session details of the users from the current user’s client with details including username, logintime, logouttime, workingminutes, ip address details.***

***5.2) Daily Print Transactions – This component is responsible for displaying the structured documents which are binded with the data for the variables present inside the document template from xml file and being downloaded on the current date. This component is used only for Print Process which is from the old part of DC.***

***5.3) Print File Export -***

***5.4) Overall Document Status – This component is responsible for displaying 5 functions. Let me explain each and every function’s functionalities.***

***5.4.1) Overall Documents – Will display all documents present inside the current user’s client as a table with details including system name, doccode, businessunit, status of the document. Apart from this it also have a filtering option to filter document’s based on the same. In addition to that you can see the history of the document about how this document is created to moved to library based on the status of the document.***

***5.4.2) Businessunit vs Document Report – This function will display a pie chart visualizing Number of documents in each businessunit categorized by the status of the documents and you can filter these documents based on Businessunit & Doccode.***

***5.4.3) User vs Documents Report – This function is responible for displaying a bar graph with data of Number of Documents created by each user and helps us to map the proportion between users and documents.***

***5.4.4) User vs Documents (Status Oriented) Report – This function is also works like the above function but instead of giving the overall document count of each user. It gives a document count of each users grouped based on document status.***

***5.4.5) Businessunit vs Comment Report – This function is responible for displaying the overall comment’s count from the documents with each category. And you can filter the data using businessunit data to view specific comment’s count from the businessunits.***

***6) Admin:***

***This module is responsible for doing all the works that will done only by Admin previleaged users and the functionalities inside this module will something like creating and managing users, creating and managing businessunit, and assigning and deassigning documents across multiple businessunits. Now I will list out each and every components inside this module.***

***6.1) Manage User – This component is responsible for managing the users of current user’s client by creating, displaying and modifying the user details. And it also have a filtering functionality for users.***

***6.2) Manage Businessunit - This component is responsible for managing the businessunit details of the current user’s client by displaying and creating businessunit. And it also have a filtering functionality for businessunit.***

***6.3) Manage Document Code – This component is responsible for managing the document code details of the current user’s client by displaying and creating document code. And it also have a filtering functionality for document code.***

***6.4) Common Doc to Bu – This component is responsible for moving library documents from ACCC businessunit to other businessunits that we have inside our current user’s client. You can also download the document’s content that are listed.***

***6.5) Bulk Doc Assign – This component is responsible for assigning library documents from “STD, STD OP” businessunits to other businessunits. And this component is only used for the users who are from “sps” client with id 104.***

***6.6) Bulk Doc Deassign – This component is responsible for deassigning the library documents which we are assigned through Bulk Doc Assign component. And this component is also restricted to users who are from “sps” client.***

***6.7) Virtual Groupdoc Setup -***

***6.8) Staging to Production – This component is responible for moving library documents from one environment from another environment likely from staging to production. And the whole data required for the library documents including businessunit details, doccode details, useraccount details is being transferred to another database in the production environment.***

***7) My Profile:***

***This module act as a profile page of our system where the user’s can see their profile information, security questions and also can edit the same as per their need. Also provides a functionality to change the password. So now lets break down each and every components of this module.***

***7.1) Personal Information – This component is responsible for displaying the user’s details including profile picture, firstname, lastname, email, nickname.***

***7.2) Address – This component allows the user to see their address details and update the same if required. And also consists a functionality to change the password by navigating the password change component.***

***7.3) Security Questions – This component is responsible for visualizing and updating the security questions of the current user. And also provides a way to view the events of users like success-login, invalid password attempt.., happened inside our system by clicking on the audit report button.***

***8) DataDictionary:***

***Up until now we have seen the all other 7 modules inside our DC application at a high level. For the last two modules namely DataDictionary, Inbox I will give you the extended explonation as because these are the ones that I have worked end-to-end.***

***This module is responsible for creating and maintaining so called variables that will be used inside structured document in XML format. If you don’t have a clear picture about the variables, you will find what it is and why it’s used inside our system after completed reading this module.***

***8.1) Tables Involved:***

***However this module is going to store and maintain only variables data, we could have gone with a single table to store the variable data alone. But inorder to maintain a structured way of storing this variables, we have created three tables to store the data effectively. These tables are namely***

***1) DataDictionary Category – This table acts as a parent for item table, as the items are created under specific categories.***

***2) DataDictionary Item – This table acts like a parent for property table, as the properties are added under specific items.***

***3) DataDictionary Property – This table is where our variables are stored.***

***So the overall table relationship / hierarchy will looks something like below.***

***DataDictionary Category***

***DataDictionary Item***

***DataDictionary Property***

***So the above will be tables and their hierarchy for our datadictionary module. And each tables stores their data and their parent table’s foreignkey id if the table has a parent table.***

***8.2) Components:***

***As we seen in the table structure above, our variables are stored and structured across three connected tables namely DataDictionary Category, DataDictionary Item, DataDictionary Property. Due to this, our datadictionary module is responsible for maintaining data across these three tables.***

***And about the components inside our module, as we are maintaining three tables, our components are also three with each component is responsible for each table. Let me break down what each component is for and the functionalities that each component have.***

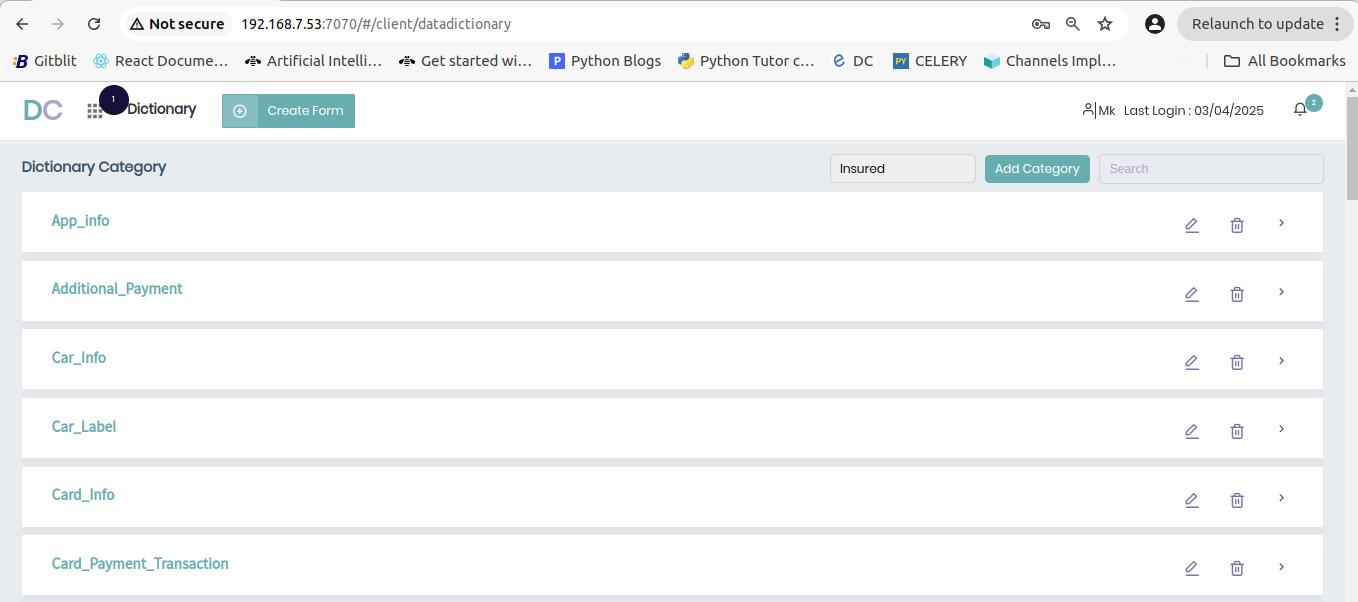
***8.2.1) Category – This component is resposible for displaying the datadictionary categories of the user by filtering the data with their client from datadictionary category table. And this component doesn’t have any options to add / delete /edit a category through UI.***

***8.2.2) Item – This component is responsible for maintaining the datadictionary items by displaying, creating, editing, deleting datadictionary items of the user by filtering the data by both their client id and the category that we choosed. This component also have search option to search datadictionary items by their name.***

***8.2.3) Property – This component is responsible for maintaining the datadictionary properties by displaying, creating, editing, deleting datadictionary properties which is contains the data that we use as variables in structured documents.***

***Now let’s see each component is displayed into our DC application. To see this, after logged in to the system, navigate to the mentioned url “****[http://192.168.7.53:7070/#/client/datadictionary](http://192.168.7.53:7070/" \l "/client/datadictionary)****” the initial IP details may vary but url is same. When hitting this url you will find a page that i attached below.***

***The All-In-One Component:***



***This page alone is maintaining all our components inside it. Let me give separate images for each component.***

***8.2.1) Category Component:***

***As mentioned in the arrow mark above, our category component is only for to display the datadictionary categories available to the current user by filtering it with their client id. And it doesn’t have any add / edit / delete functionalities to do from the UI.***

***And the API that fetches the data for this displaying functionality of this Category Component is given below.***

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary/getcategory/***](http://192.168.7.53:5464/Datadictinary/getcategory/)

***API Functionality: This API is responsible for fetching the datadictionary categories of the users from the Datadictionary Category table by applying client filter to it.***

***8.2.2) Item Component:***

 ***Display Items Add Items Search***

***Edit Item Delete Item***

***As mentioned in the above item component, we have a complete crud operations for the Item component and let me break down each of their function.***

***Item Display Function:***

***This function is responsible for listing all the Items that are present in the user’s client and selected category in category dropdown (in the image the category is “Insured”). So we can see the API description below. And there is also a hierarchy called subitem which comes in between Item and Property. However there is no option to create a sub item inside the UI, but we have some subitem entries in the database.***

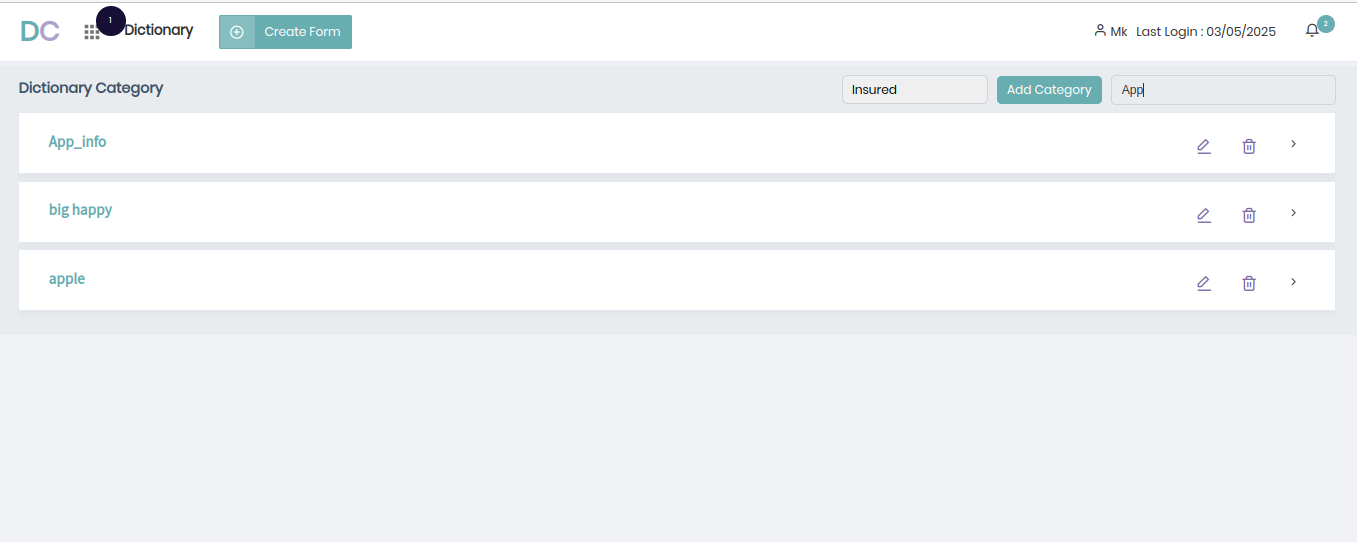
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary/getitem1/***](http://192.168.7.53:5464/Datadictinary/getcategory/)

***API Functionality: This API is responsible for fetching the datadictionary items of the users from the DatadictionaryItem table. And this receives category id as it’s input and does the filtering with both category and client id and return the datadictionary items for display.***

***Item Filter Function:***

***This function is used filter datadictionary Items by their name and display the matched items from the item list.***

******

***As you can see above in the picture we typed “App” into the search bar and the function only returns the items whose name contains the given characters / word. However this functionality is handled by frontend alone, so that we didn’t used any APIs for this functionality.***

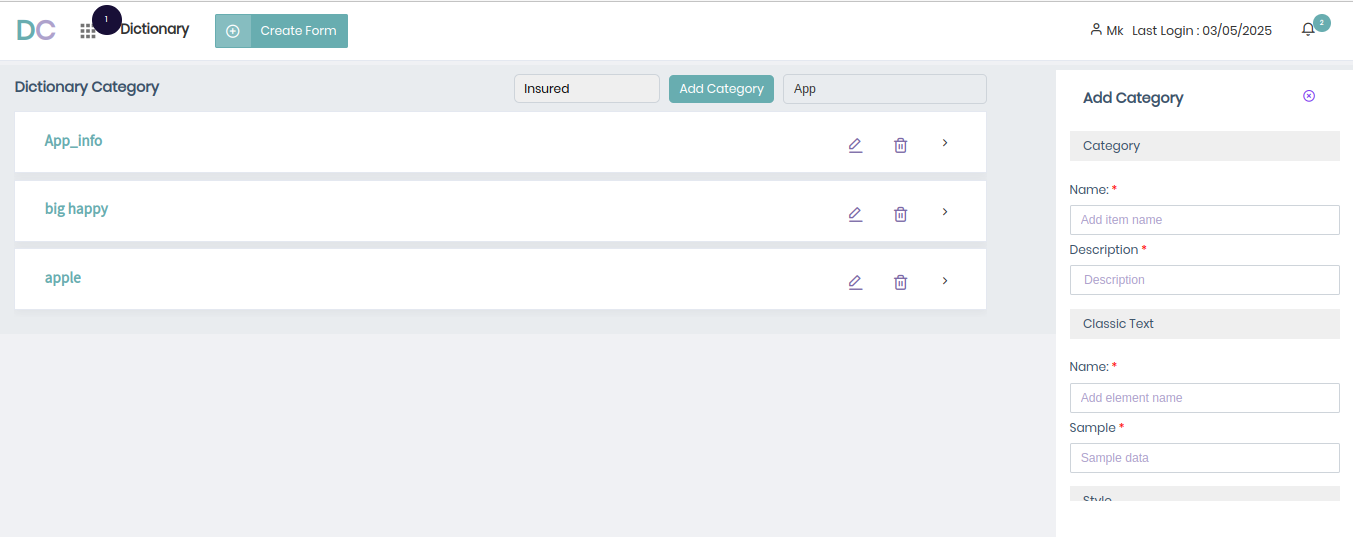
***Add Item Function:***

***This function is responsible for adding a new item under the specific category. While adding a item to the category, we must need to add a property to that Item also which to ensure there is no item will be added without a single property. And on clicking on the “ADD Category” button above you will see a form to fill the item and property details and submitting the same will call a API to create the entries inside the database.***

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary/additem***](http://192.168.7.53:5464/Datadictinary/additem)

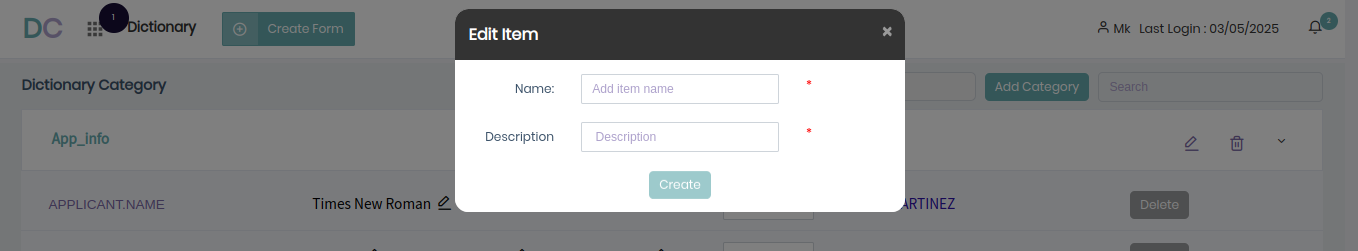
***API Functionality: This API is responsible for creating the new item under a given category and creating a new property under the newly created item. For doing this it receives category, item and property details as input and creates the same in relevant tables.***



***As marked by arrow above, this is the form that will open when you clicking on the “Add Category” button which opens the form and allows us to create both item and property.***

***Edit Item Function:***

***This function is resposible for editing the item details mainly the item’s name and description. And on clicking on the “Edit Icon” inside the Item card, you will have a pop up to you by using which you can edit the Item’s informations.***



***As marked in the arrow above, clicking on this Icon will open up a pop up with form fields, to allow users to edit the Item. And for doing this we use a API which is explained below.***

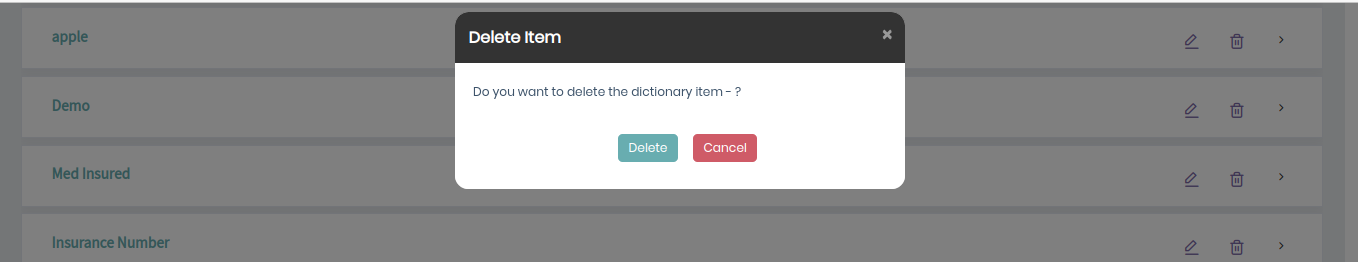
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary/edititem***](http://192.168.7.53:5464/Datadictinary/edititem)

***API Functionality: This API is responsible for editing the specific items details that we want to edit. It receives Name, Id, Description of the Item and input and update the Item details with relevant item id.***

***Delete Item Function:***

***This function is responsible for deleting an item from the category and this can be done by clicking on the delete icon and this will open up a pop up which is asking for confirmation. And delete the item upon confirmation. For doing it is calling a API which will be mentioned below.***



***As you can see in the above image, on clicking on the delete button on the Item card will open up a pop up for asking confirmation to delete and doing the confirmation will call the API to delete the item.***

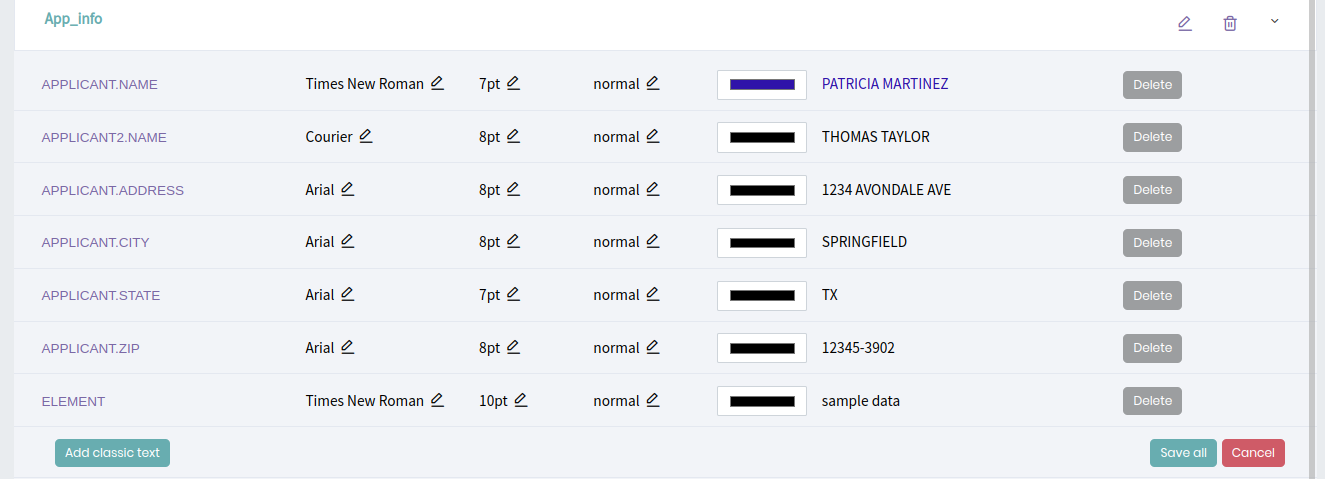
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary/deleteitem1***](http://192.168.7.53:5464/Datadictinary/deleteitem1)***/***

***API Functionality: This API is responsible for deleting a item from the database. To do that it receives item’s id as input and delete the item with relevant id.***

***8.2.3) Property Component:***

***Display Property Delete Property***



***Add Property Edit Property***

***As you can see in the above image, This module is responsible for managing the datadictionary property by displaying, creating, editing, removing properties under specific Items. Let me break down each and every functionalities of the property component.***

***Property Display Function:***

***As you can see in the above image, properties are displayed for the specific Item. And you can click on other Item cards to view what are the properties are found under the specific clicked item. For displaying the properties relevant to the item, we use an API which is mentioned below.***

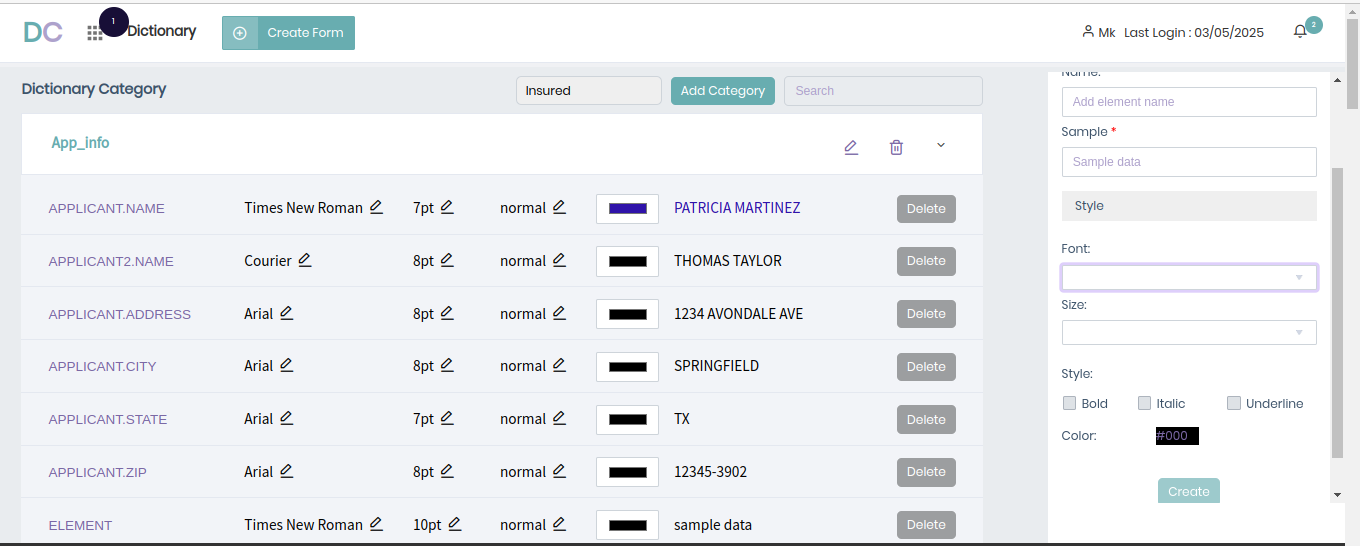
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary/getproperty1***](http://192.168.7.53:5464/Datadictinary/getproperty1/)

***API Functionality: This API is responsible for fetching the property data for the given item from the database. To do that it receives item’s id as input and retrive all the properties available under the given item.***

***Add Property Function:***

***This function is responsible for adding a new property below specific item. And you can create a new property by clicking on the “add classic text” button which is found at the left cornor of the component. And this will opens a pop up which is a form and collects the details required for adding a property and adds the same to the database by calling the API which is mentioned below.***



***As you see in the above component, the form in the right corner will be opened when clicking on the “add classic text”. And after filled the details and clicking the create button will call the API to create the property.***

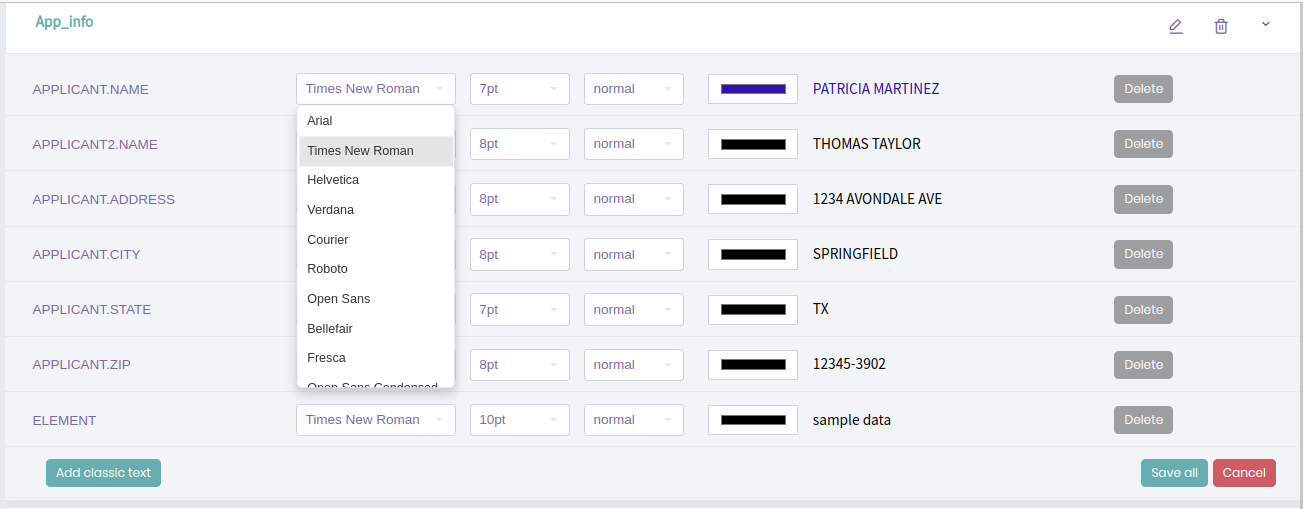
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary***](http://192.168.7.53:5464/Datadictinary/getproperty1/)***/addproperty***

***API Functionality: This API is responsible for creating a new property into the selected item and To do this it receives propertyname, description, font family, font size, font style and font color as input and creates a new property after checking the new property name is not already found inside the system.***

***Edit Property Function:***

***This function is responsible for editing the property details for all the properties available under specific item. You don’t need to update each property one by one, You can edit all the available properties inside the item that you want to update and save the changes to the database by calling the below mentioned API.***



***As you can see in the above image, you can edit multiple properties at same time and save the same.***

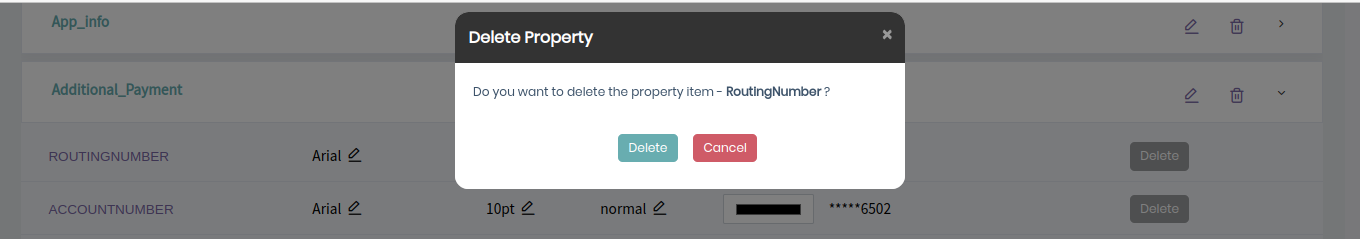
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary***](http://192.168.7.53:5464/Datadictinary/getproperty1/)***/editproperty1***

***API Functionality: This API is responsible for updating multiple property details of specific item. To do this it receives the all updated details of the property and updates the same inside the database.***

***Delete Property Function:***

***This function is responsible for deleting a property from the specified Item. And this function can be triggered when clicking on the delete button will opens up a popup that will ask for confirmation and deletes the selected property upon confirmation by calling the below mentioned API.***



***APIs:***

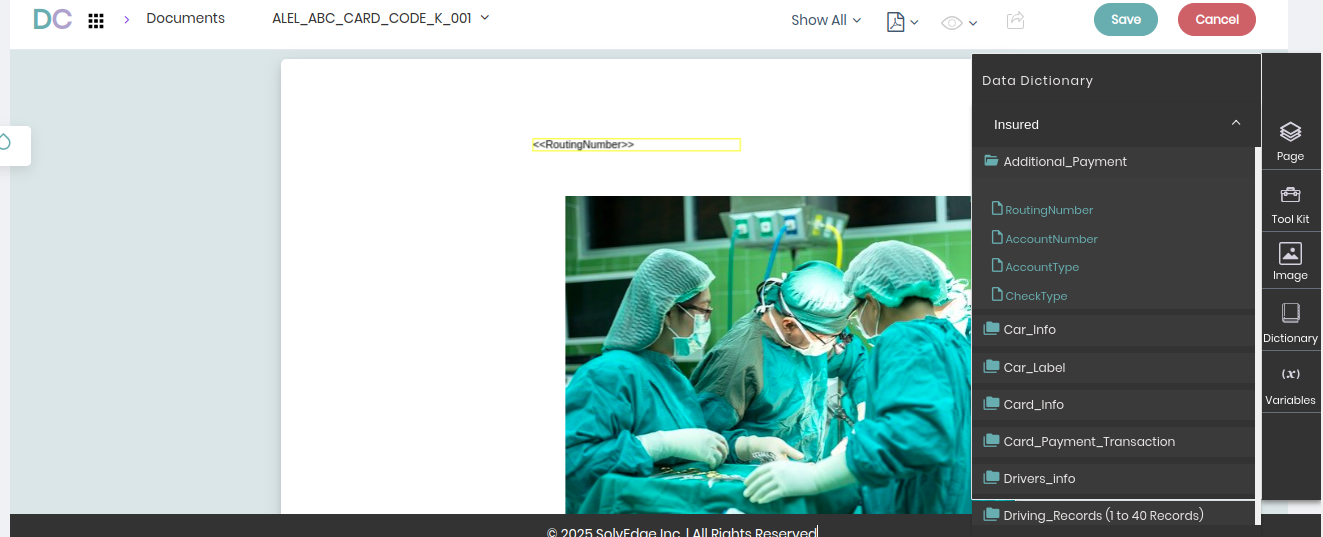
***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary***](http://192.168.7.53:5464/Datadictinary/getproperty1/)***/deleteproperty1***

***API Functionality: This API is responsible for deleting specific property from the database. Receives the property id as input and deletes the same from the database.***

***Up untill now, we have seen how we are maintaining the category, item, property details like creating / editing / deleting the same. Now we are heading into section where we will use this variables which will be our structured documents. Will see them now!***

***Variables Usage In Structured Document Component:***

***In this component we use all our properties as varibles that we created and maintained in datadictionary module. To retrieve the variables data into this component we use the below mentioned API.***



***From the above image you will get to know how we can use the variables that we have created inside our datadictionary module.***

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Datadictinary***](http://192.168.7.53:5464/Datadictinary/getproperty1/)***/getitem***

***API Functionality: This API is responsible for fetching all the categories and their items and their relevant properties in one API to visualize the same.***

***9) Inbox:***

***This module is responsible to move a document from it’s initial status which is “In Progress” to “Library” status which is obtained when the reviewer approved the document. This module acts as a emailing layer between the designer and reviewer to effectively send messages with documents for approvals to make the document move from one status to another status until it reaches “Library” status. Apart from this, this module also contains a messaging interface for per document communications between designer and reviewer.***

***Inside this Inbox module it has various categories which categorize the message that is being sent and received by the users into Inbox, Archieve, Review, Pending, Approved, Library, Sent.***

***To get a clear understanding about how the document is being pushed to the library via our Inbox system, Let me a sample flow of the same.***

***Inbox Normal Flow:***

***1) First of all, A designer creates a new document with the relevant contents and saves the document current the newly created document is in “In Progress” status. Once they completed the document, They put a mail to a reviewer with the document attached and asking for approval for the same. After submitting the document for approval, it’s status will changed to “Submitted”.***

***2) So after submitting the document to a reviewer the designer waits for the reviewer’s approval. In the meantime the corresponding reviewer receives the message that the designer send to them via Inbox. And then he reviews the document for correctness.***

***2.1) If everything seems to be right inside the document, the reviewer approves the document as send a email message for the same. Now after sucessfull approval the document get “Approved” status.***

***2.2)If there are any changes found inside document, the reviewer put a comment indicating the changes that he want to make in the document and revert the message to the designer of the document by assigning the document to reedit. Now in this phase the document will be in “Re Edit” status. And again the designer need to done the changes and assign it for approval and the flows go on until the reviewer approves the document.***

***Inbox Admin Flow (Move to Staging):***

***In this flow, instead of the designer the admin creates the document and with right document content with their own responsibility. Here the admin have two options***

***1) Either they can follow the normal flow that stated above by sending the document to reviewer for approval.***

***2) Or there is a option called “Move to Staging” by which the admin can directly push the document to “Library” status.***

***So this is how the inbox flow looks like now will see how it actually looks like inside our DC application after getting known what are components and relevant functionalities of our module.***

***9.1) Tables Involved:***

***As previously seen in the Inbox Module’s introduction we will be mainly operating with mainly two components including Inbox and Comment., Additionally Chat Component. So Respective to each component we have three tables involved which are listed below.***

***9.1.1) Inbox – This table is responsible for maintaining the Inbox messages that has been sent and received by the users.***

***9.1.2) Comment, CommentMessage – These tables are responsible for maintaining the comments that reviewer do on the documents and the replys that user do for the comments are stored.***

***9.1.3) ChatMessage – These table is responsible for storing the messages that the designers / reviewers do on the per document basis.***

***9.2) Components:***

***As our module requires multiple inter connected components to work including document editor component, view document component.., We would mainly focus on the component which is ultimately defined within Inbox Module. With this being the situation, we have three components inside this module.***

***9.2.1) Inbox Display Component – This component is responsible for maintaining the messages by categorizing them into categories like Inbox, Archieve, Reedit, Pending, Library, Sent by allowing Display, Create, Filter options on the inbox messages.***

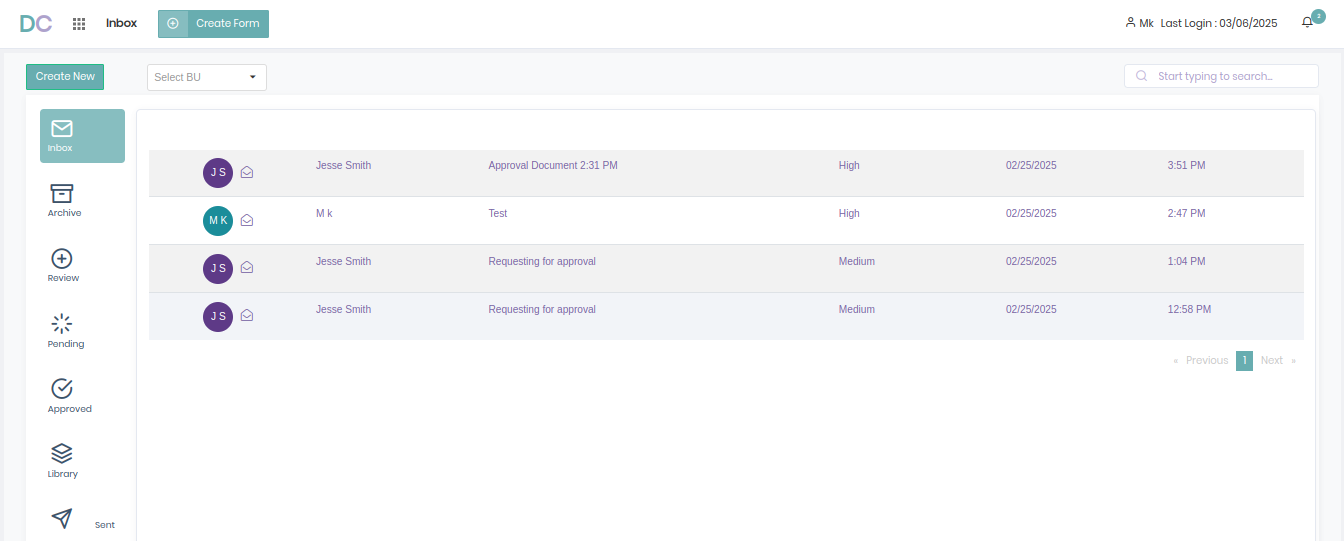
***9.2.2) Chat Message Component – This component is part of our Inbox component, Which is responsible for maintaining the chats that user makes on a per document basis.***

***9.2.3) Comment Component – However this is not a main component, which is used as a sub component inside the document viewer page. This component is responsible for maintaining comments of the users about the document with functions including display, create, edit, delete and filtering. This component not only has comment component but also has a reply component also to create reply for the comments that user made.***

***Now let me eloborate on each of the components and their functionalities.***

***9.2.1) Inbox Component:***

***This component is responsible for displaying the Inbox messages that the user have based on the available categories including Inbox, Archieve, Reedit, Pending, Library, Sent. And also have create and filtering functions to them. Let me explain each and every functionalities of the same. To create a Inbox message you can click on the “Create New” button or You can click the share icon present inside the document editor.***

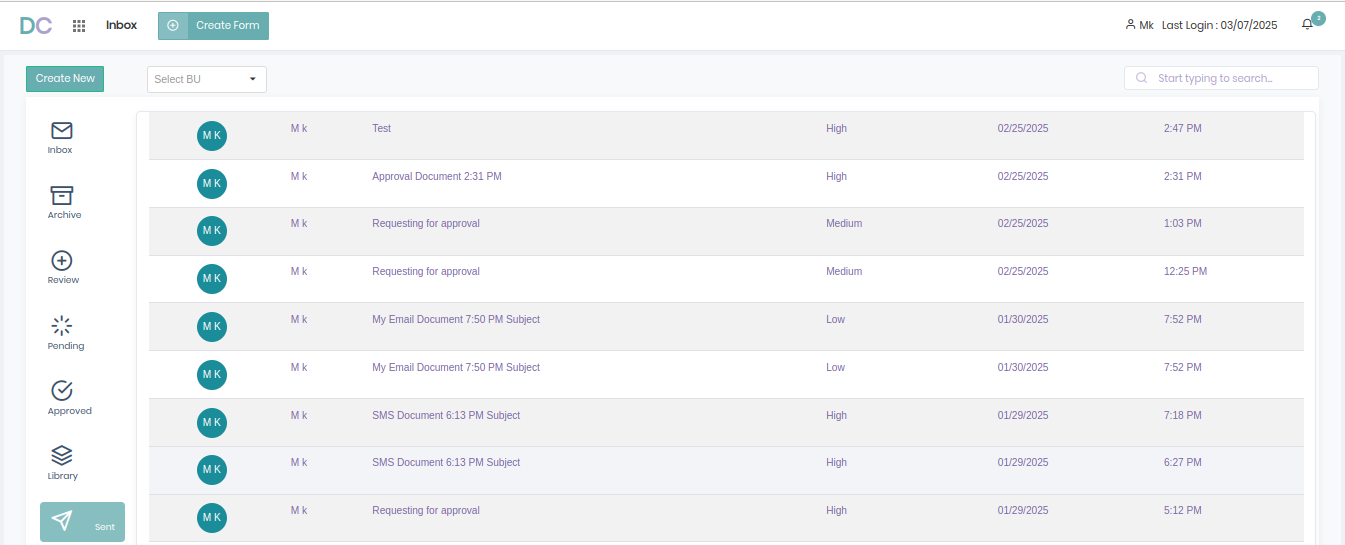
 ***Filter Message (BU) Display Message(Inbox)***

***Create Message Filter Message(Subject)***

***As you can see above we have multiple functionalities inside our Inbox Module, Let me break each of them in detail in upcoming parts.***

***Display Inbox Message:***

***This function is responsible for displaying the inbox message of the users based on the category including Inbox, Archieve, Review, Pending, Approved, Library, Sent with each category having specific conditions to return the relevant messages. However all the categories is going to display the list of messages in some way, we could have use only one API which receives the category as input and returns the relevant data. But however the current system has Two APIs to handle the display function.***



***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/totalmessage***](http://192.168.7.53:5464/Inbox/totalmessage)

***API Functionality: This API is responsible for fetching categories specific messages from Inbox module, receives the type of the Inbox, Page No as input and returns the relevant messages. This API is being called for categories except “Sent” category.***

***Inbox: Receives type (“inbox”) and pageno as input and returns the all messages received by the user in last 30 days based on pageno.***

***Archieve: Receives type (“Archived”) and pageno as input and returns the all messages received by the user before last 30 days based on pageno.***

***Review: Receives type (“ReEdit”) and pageno as input and returns the messages in which a user’s document is re-assigned for changes based on pageno.***

***Pending: Receives type (“Approval”) and pageno as input and returns the messages in which a other user’s document is came for the user’s approval based on pageno.***

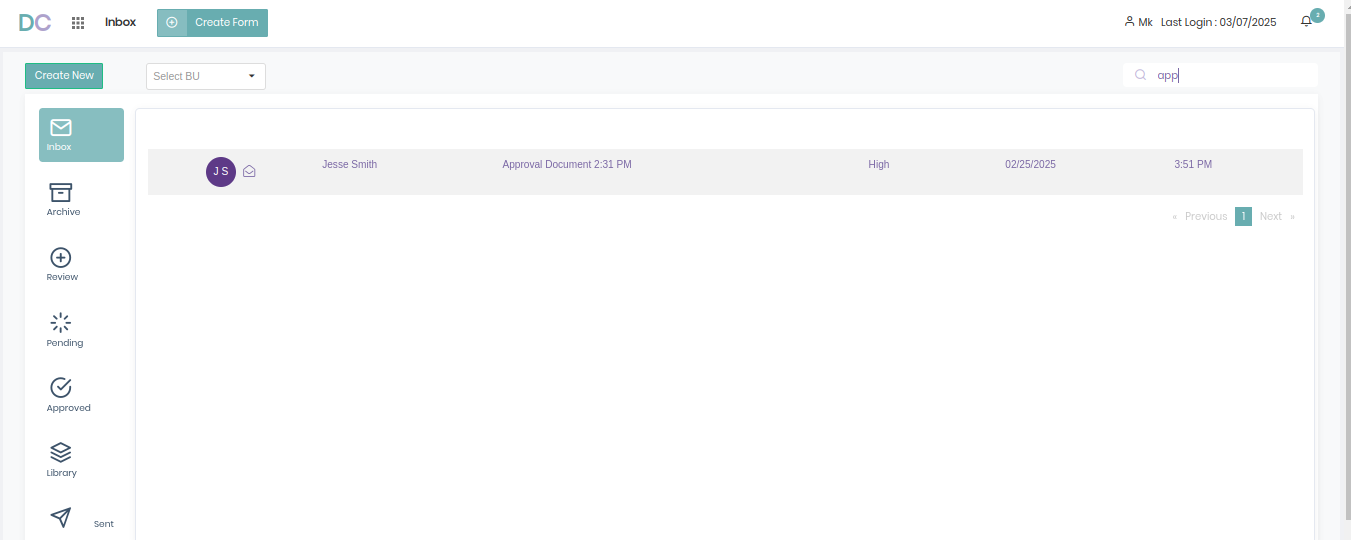
***Library: Receives type (“Allotment”) and pageno as input and returns*** ***messages in which the user’s document are got approval to move to the library based on pageno.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/sentmessage***](http://192.168.7.53:5464/Inbox/totalmessage)

***API Functionality: This API is responsible for fetching messages which are sent by the user and only being called while clicking on “Sent” category of Inbox as pictured in the above message.***

***Search Message Function:***

***This function is responsible for searching the inbox messages based on their subject and return matched messages based on category types. It receives the search text as input and returns the filtered messages as output.***

******

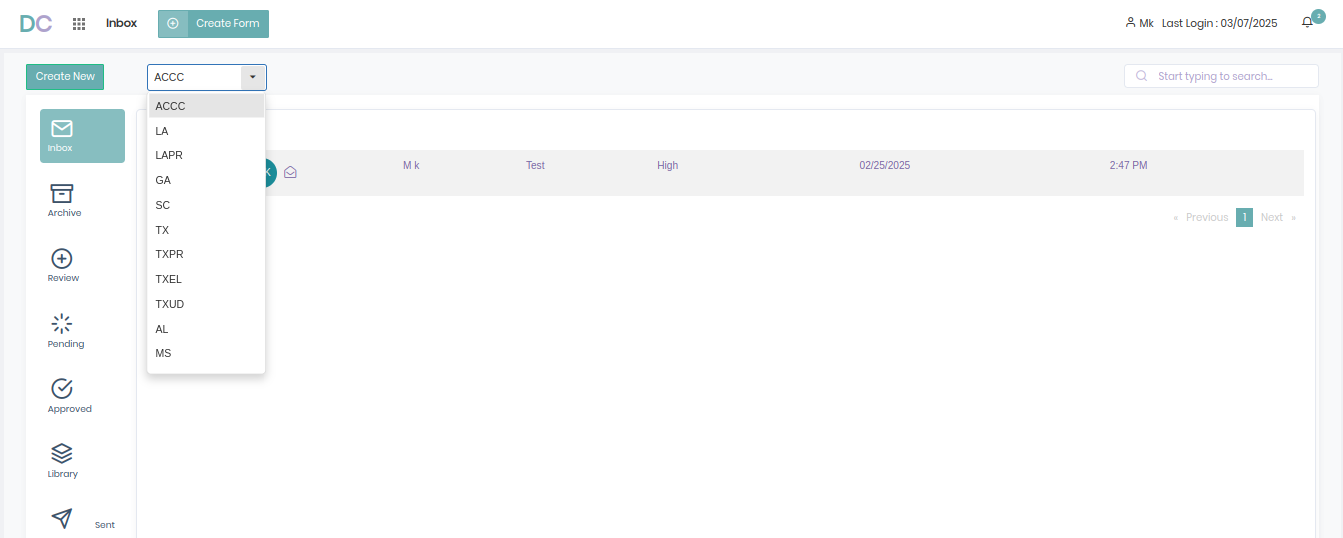
***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/searchfilter***](http://192.168.7.53:5464/Inbox/searchfilter)***/***

***API Functionality: This API is responsible for search through the inbox messages based on the message’s subject and return messages that matches with the search query.***

***Filter (BU) Message Function:***

***This function is responsible for filtering the messages based on the their document’s businessunits. To filter only messages of the documents inside the given businessunit. Receives Businessunit Id as input and return the relevant messages as output.***

******

***As you can see in the image above, we have filtered the Inbox messages into only messages that are from document’s Inside the selected businessunit “ACCC” here.***

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/bulistfilter/***](http://192.168.7.53:5464/Inbox/bulistfilter/)

***API Functionality: This API is responsible for filter out the messages inside any Inbox category based on the Businessunit Selected. Receives businessunit id and pageno as input and return filtered messages from Inbox Table.***

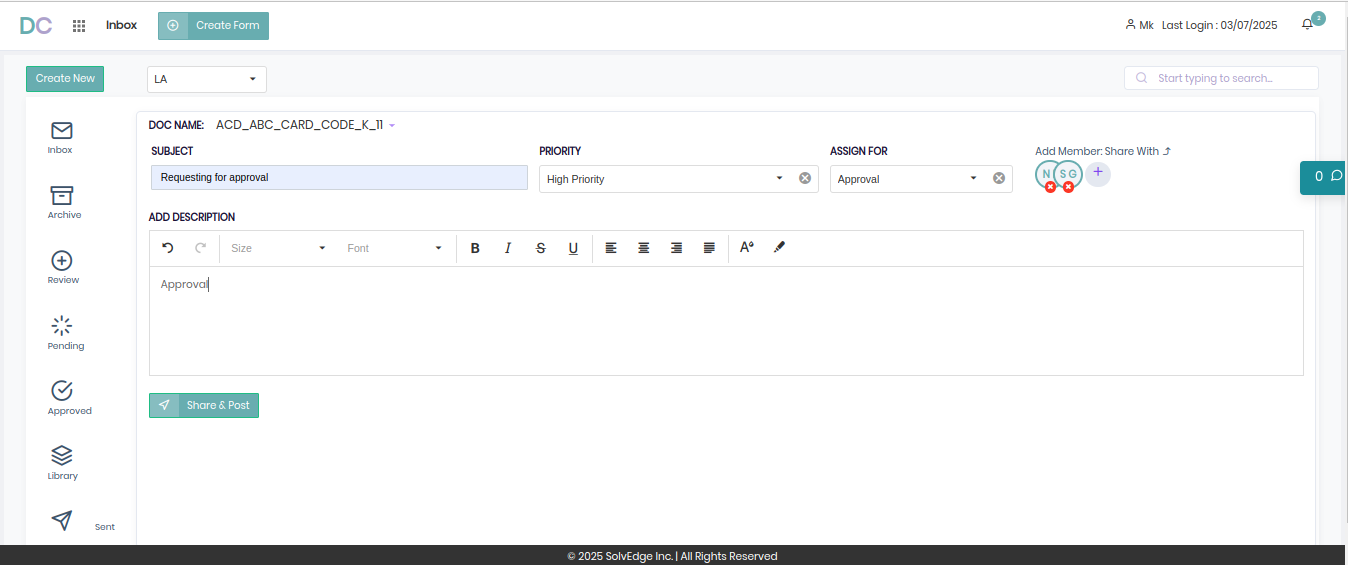
***Create Message Function:***

***This function is responsible for creating a new inbox message to a document whether sending the document to approval, Reedit, Move to staging. And you can reach Create New Message component via two ways***

***1) Clicking on the “Create New” Button***

***2) By clicking on the Share Icon present in document view component.***

***After reaching the component you will fill up the details like Subject, Priority, Receivers of the message and Body of the message. After filled up the details, you will click the “Share & Post” button below to sent the message which eventually creates a entry inside the Inbox Table.***



***APIs:***

***There will be two APIs called for creating a Inbox entry. One from Designer / Admin side called “inboxvalues”, Another one from Reviewer side called “reeditsaving”. Along with this we also have 2 more APIs inside this component which are***

***“CommentList”, “PrepopulateList” - These APIs are used to fetch the comments & previous message details of the document.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/inboxvalues***](http://192.168.7.53:5464/Inbox/inboxvalues/)

***API Functionality: This API is responsible for senting a new message to Reviewer / Admin for approving the document. For doing this it receives the document details, message’s subject, description, receivers details as input and creates a entry in Inbox table.***

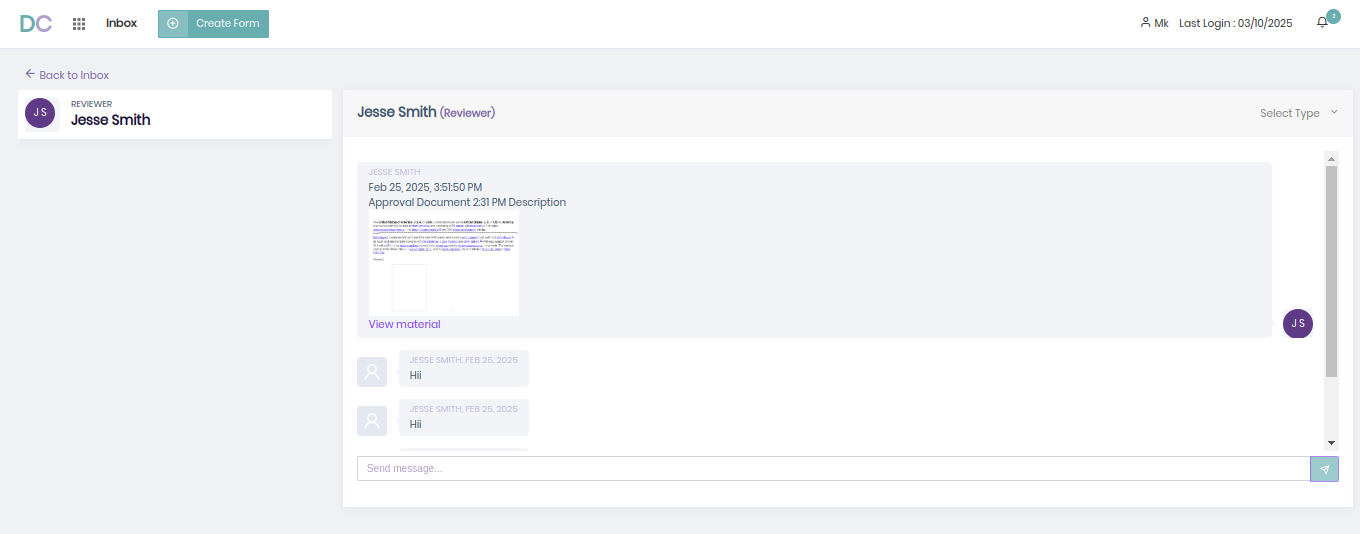
***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/reeditsaving/***](http://192.168.7.53:5464/Inbox/reeditsaving/)

***API Functionality: This API is responsible for senting a reply inbox message to the Admin / Desinger who created the document about it’s approval status whether it’s “Approved / Re Edit”. And it’s most probably used in the reviewer side collecting only assigned for as input data and use other input required for the Inbox table from the previous approval message that is sent by the Designer.***

***9.2.2) View Inbox Message / ChatMessage Component:***

***This component is used to view the detailed version of our inbox messages including our document’s content and sender’s details, subject, description details. This component can be rendered when we click on the Inbox Message that we previously saw inside the list in Display message component. And this is the component, In which our chat messaging component is integrated on.***

***As this component consists of multiple functionalities of display Inbox Message and also Create and Display ChatMessages that the users communicate with, let me break down each and every of them below. Before that I have given the overall component’s Image below.***



***Display Inbox Message Function:***

***This function is responsible for displaying the detailed version and information of the inbox message along with the document’s image with a reference to the actual document’s view page.***

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/userdetail/***](http://192.168.7.53:5464/Inbox/userdetail/)

***API Functionality: This API is responsible for fetching the complete information about the inbox message including subject, description, document’s details and also the chatmessages for the inbox message. It receives the inbox id as input and returns the relevant details of the inbox entry including chatmessage details, Docuemnt Image and Inbox Subject, descrtption, created\_date.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/messagedetail/***](http://192.168.7.53:5464/Inbox/messagedetail/)

***API Functionality: This API is responsible for creating a new chatmessage into the table for the given inbox message by collecting document id, message, fromuser and touser as input.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/userdetailnostatus/***](http://192.168.7.53:5464/Inbox/userdetailnostatus/)

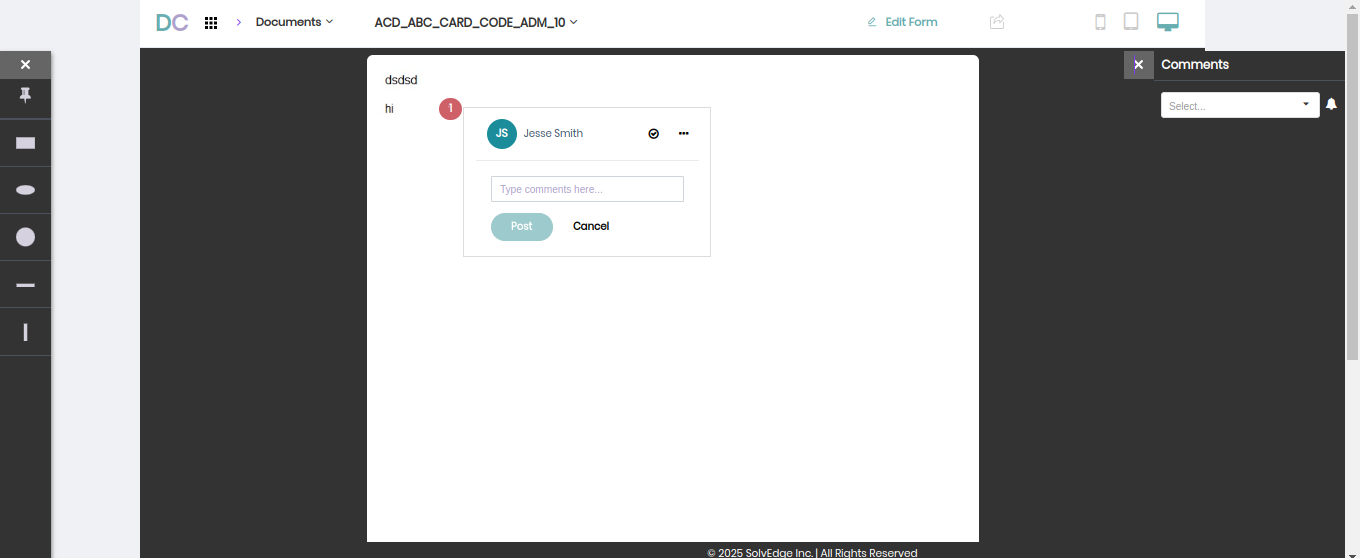
***API Functionality: This API is responsible for fetching the updated information of the inbox message like the “userdetail” API do. But the main difference here is that It’s only called to fetch the newly created chatmessage data to the UI.***

***These are the components that we majorly use on Designer / Reviewer side of our Application Especially on Inbox Module. Now let see the remaining one component which is mostly used by Reviewer which is “Comment Component”.***

***9.2.3) Comment Component:***

***This component is mainly used by Reviewers to manage comments for each documents that came to them for approval by add / delete / edit / get /resolve comments to indicate the changes that the reviewer want to see inside the document. This component is rendered when the Reviewer clicks on the View Material Link provided in the View Message Compoent and redirected to the View Document Compoent where our Comment Component is Integrated.***

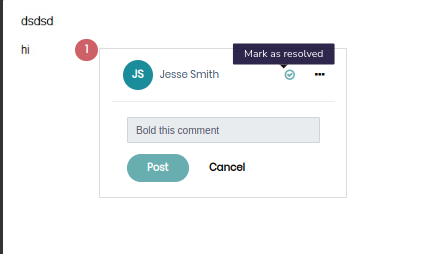
***As this component has multiple functionalities including get / add / edit / delete / update / filter on comments, Let me break down each of them one by one.***

******

***As You can see in the above image we can see the Comment Sub Component, On our Document Display Main component. Let me list out their functionalities onr by one below.***

***Add / View Comment Function:***

***This function allow you to create & view your comment with a form like structure this form also have options to resolve, edit and delete the comment.***



***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/getcomment***](http://192.168.7.53:5464/Inbox/getcomment)

***API Functionality: This API is responsible for fetching the content of the comment of the given comment is. It receives the comment id as input and fetch the content of the comment from the db.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/createcomment***](http://192.168.7.53:5464/Inbox/createcomment/)

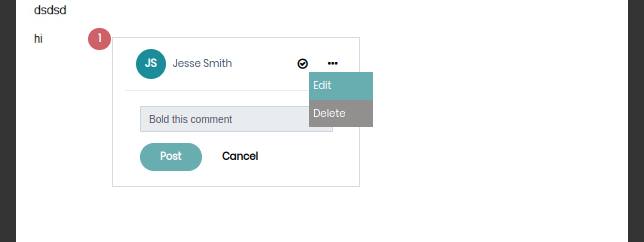
***API Functionality: This API is responsible for creating a new comment message for the document which receives the document id, userid, comment details as a input to create the same in the database.***

***API Endpoint: http://192.168.7.53:5464/Inbox/markedresolved***

***API Functionality: This API is responsible for marking the comment as resolved means that the change that given in the comment is done. And this API receives the comment id as input and make the comment as resolved by updating it’s status.***

***Edit / Delete Comment Function:***

***These two functions are responsible for providing editing / deleting a comment. And receives commentid as input for delete, commentid and updated comment details as input and does the deletion and updation on comment table.***



***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/***](http://192.168.7.53:5464/Inbox/getcomment)***editcomment***

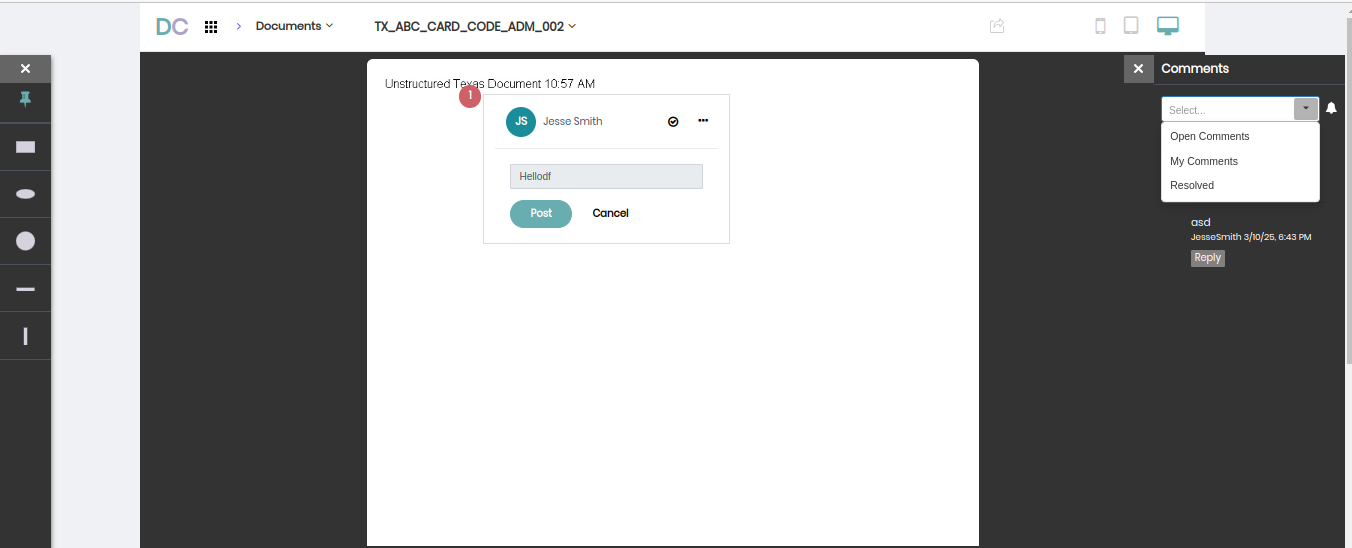
***API Functionality: This API is responsible for editing the existing comment details by receiving comment’s content, designproductid as inputs and edits the data from db.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/***](http://192.168.7.53:5464/Inbox/createcomment/)***deletecomment***

***API Functionality: This API is responsible for deleting an existing comment by receiving the comment id as input and deletes the same from the db.***

***Filter Comment Function:***

***This function is responsible for filtering the comments based on three categories including “Open Comments”, “My Comments”, “Resolved” based on their logic.***

******

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/commentfilter***](http://192.168.7.53:5464/Inbox/commentfilter)

***API Functionality: This API is responsible for filtering the comments based on the category with relevant logic.***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/***](http://192.168.7.53:5464/Inbox/commentfilter)***usercommentList***

***API Functionality: This API is responsible for fetching all the comments and replies that the document have.***

***Add Reply Function:***

***This function is responsible for adding a new reply to a existing comment by typing the reply message into a same form like comment creation form.***

***APIs:***

***API Endpoint:*** [***http://192.168.7.53:5464/Inbox/***](http://192.168.7.53:5464/Inbox/getcomment)***replycomment***

***API Functionality: This API is responsible for creating a new reply message by receiving content, designproductid, comment id and created a reply message in the for the same.***

***WITH ALL OF THE ABOVE DOCUMENTATION, YOU CAN GET AN IDEA ABOUT DC AND HOW EACH MODULE WORKS INSIDE DC. HERE I HAVE EXPLAINED THE DATADICTIONARY AND INBOX MODULE IN DETAIL REST OF THE THINGS WILL BE COVERED BY RELEVANT PEERS.***